

**CONTINUOUS AREA ALLOCATION SYSTEM**

Patent Number: JP6139703  
Publication date: 1994-05-20  
Inventor(s): MATSUMOTO SATOSHI; others: 02  
Applicant(s): HITACHI LTD  
Requested Patent: ☐ JP6139703  
Application Number: JP19920288397 19921027  
Priority Number(s):  
IPC Classification: G11B20/12; G06F12/00  
EC Classification:  
Equivalents:

---

**Abstract**

---

**PURPOSE:** To improve access performance by successively bisecting all the areas of a RAM, arranging the leading blocks of respective files at dividing points, continuing blocks to this arranged block thereafter, finding out any place when the arranging place is occupied with the same method, and continuously arranging blocks to this place thereafter.

**CONSTITUTION:** 1-1 to 1-7 shown in diagram are the leading blocks of files, and 4 is the RAM. The leading block 1-1 is arranged (2) at the position of a search pointer 2 from an initial state (1), and the pointer 2 is moved for 1/4. In the state (3) of generating two files, the pointer is moved for 1/8 so as to deal with the generation/erase of the file as prescribed. In a state (6), the pointer 2 shows 1/2 in a state (6) but since the block 1-1 is already arranged, the next 1/4 is investigated, the leading block 1-6 is arranged in the empty area, and the pointer is moved for 3/4. When the size of the file is small, continuous areas can be allocated to a lot of files. The files in a big size are allocated to several continuous areas. The number of times for the I/O of files is reduced.

---

Data supplied from the esp@cenet database - I2

(19)日本国特許庁(J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平6-139703

(43)公開日 平成 6 年(1994) 5 月20日

(51)Int.Cl.<sup>5</sup>

G 1 1 B 20/12

G 0 6 F 12/00

識別記号

5 0 1 H

庁内整理番号

9295-5D

8526-5B

F I

技術表示箇所

審査請求 未請求 請求項の数3(全 6 頁)

(21)出願番号 特願平4-288397

(22)出願日 平成 4 年(1992)10月27日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目 6 番地

(72)発明者 松本 智

横浜市戸塚区吉田町292番地株式会社日立

製作所マイクロエレクトロニクス機器開発

研究所内

(72)発明者 幡野 富久

横浜市戸塚区吉田町292番地株式会社日立

製作所マイクロエレクトロニクス機器開発

研究所内

(74)代理人 弁理士 小川 勝男

最終頁に続く

(54)【発明の名称】 連続領域アロケーション方式

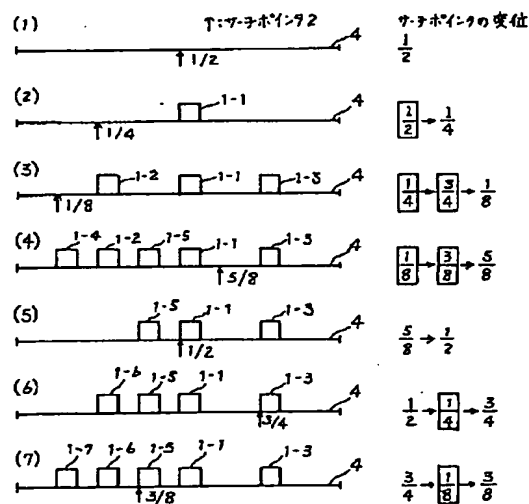
(57)【要約】

【目的】ランダムアクセス記憶装置内のファイルのフラグメンテーションを減らし、ファイルアクセス時のI/O回数を減らしてランダムアクセス記憶装置のアクセス性能を向上させること。

【構成】ランダムアクセス記憶装置を次々と二分割していき、その二分割点にファイルの先頭ブロックを割当て、ファイルの二番目以降のファイルは先頭ブロックの後方に連続して割り当てる。

【効果】隣合う二分割点の距離がファイルサイズよりも小さくなるまでファイルに連続領域を割り当てることができるため、ランダムアクセス記憶装置を効率良く使用しながらファイルアクセスを高速化できる。

図 1



は、アロケーションされたことを示す。

## 【特許請求の範囲】

【請求項1】ランダムアクセス記憶装置において、前記記憶装置の全データ領域を次々と二分割していき、前記二分割点に各々のファイルの先頭ブロックを割り当てることを特徴とする連続領域アロケーション方式。

【請求項2】前記記憶装置において、前記二分割点に割り当てられた前記先頭ブロック以降の前記ファイル中のブロックを、前記先頭ブロックの直後に連続して割り当てることを特徴とする請求項1記載の連続領域アロケーション方式。

【請求項3】前記連続して割り当てたブロックの直後のブロックが既に割当て済みだった場合、前記ファイル中の次のブロックは前記先頭ブロックを割り当てる方法に従い割当て、それ以降の前記ファイル中のブロックは前記次のブロックの直後に連続して割り当てることを特徴とする請求項1記載の連続領域アロケーション方式。

## 【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、ランダムアクセス記憶装置にデータを格納するときの、ファイルの配置方式に関する。

【0002】

【従来の技術】従来は図3に示すように、記憶装置を3-1、3-2、3-3、3-4・・・の固定サイズの連続領域に分割し、各ファイルを各連続領域に配置している。ファイル数が連続領域数を越えた場合、一つの連続領域中に複数のファイルが混在することもある。

【0003】

【発明が解決しようとする課題】従来の技術では、ファイル数が連続領域数を越えると一つの連続領域に複数のファイルが混在し、連続領域を確保しきれなくなる。そのためI/O回数が増え、ファイルアクセスの性能が劣化する。

【0004】

【課題を解決するための手段】上記課題を解決するために、ランダムアクセス記憶装置の全領域を次々と二分割していき、各々の二分割点に各ファイルの先頭ブロックを配置する。また、先頭ブロックの次以降のブロックは、先頭ブロックに連続するように配置する。さらに、連続するように配置しようとするブロックの配置場所が使用中の場合、そのブロックの配置場所は先頭ブロックの配置場所を見つけないと同じ方法で見つける。それ以後のブロックはそのブロックに連続するように配置する。

【0005】

【作用】ファイルが生成される度に連続領域が割り当てられるため、一つの連続領域に複数のファイルが混在することが無くなる。しかし、ファイル数が増えてくると連続領域の長さが短くなっていく。が、一つの連続領域に入りきらなかった部分は別の連続領域に配置されるた

め、それほどフラグメンテーションは激しくならない。

【0006】

【実施例】本発明の一実施例を、図を用いて説明する。

【0007】まず初めに、本発明の考え方について説明する。

【0008】図1は本発明におけるファイルの先頭ブロックを配置していく過程を示した図である。1-1～1-7は各ファイルの先頭ブロックを示し、2はサーチポインタを示し、4はランダムアクセス記憶装置を示す。

10 左側の図は記憶装置4中のどの部分にどのファイルの先頭ブロックが割り当てられたかを示し、右側の図はサーチポインタ2の変位を示している。

【0009】(1)は初期化状態であり、記憶装置4上には一つもブロックが無く、サーチポインタは1/2の部分を示している。

【0010】(2)は、(1)の初期化状態からファイルの一つ生成した状態を示す。ファイルの先頭ブロック1-1はサーチポインタ2が指し示していたところに配置され、その後、サーチポインタ2は1/4のところへ移動する。

【0011】(3)は、(2)の状態からファイルを二つ生成した状態を示す。まず、一つ目のファイルの先頭ブロック1-2をサーチポインタ2が指し示している1/4のところへ配置し、サーチポインタ2を3/4のところへ移動する。次に、二つ目のファイルの先頭ブロック1-3をサーチポインタ2が指し示している3/4のところへ配置し、サーチポインタ2を1/8のところへ移動する。

30 【0012】(4)は、(3)の状態からさらにファイルを二つ生成した状態を示す。まず、一つ目のファイルの先頭ブロック1-4をサーチポインタ2が指し示している1/8のところへ配置し、サーチポインタ2を3/8のところへ移動する。次に、二つ目のファイルの先頭ブロック1-5をサーチポインタ2が指し示している3/8のところへ配置し、サーチポインタ2を5/8のところへ移動する。

【0013】(5)は、(4)の状態からファイルを二つ削除した状態を示す。まず、ブロック1-4を含むファイルを削除すると、サーチポインタ2は1/2にリセットされる。次に、ブロック1-2を含むファイルを削除すると、サーチポインタ2は再び1/2にリセットされる。

【0014】(6)は、(5)の状態からファイルを一つ生成した状態を示す。サーチポインタ2は1/2の部分の指し示しているが、そこには既にブロック1-1が配置されているため、次の場所の1/4の部分をチェックする。そこはこの場合空き領域なので、そこにファイルの先頭ブロック1-6を配置し、サーチポインタ2を3/4に移動する。

50 【0015】(7)は、(6)の状態からファイルを一つ

生成した状態を示す。サーチポインタ2は3/4の部分  
を指し示しているが、そこには既にブロック1-3が配  
置されているため、次の場所の1/8の部分をチェック  
する。そこはこの場合空き領域なので、そこにファイル  
の先頭ブロック1-7を配置し、サーチポインタ2を3  
/8に移動する。

【0016】次に、ファイルの先頭のブロック以降のブ  
ロックの配置方法を図2を用いて説明する。

【0017】1-1-1, 1-1-2, 1-1-3, 1-  
1-4, 1-1-5, 1-1-6, 1-1-7, 1-  
1-8は、あるファイルを構成するブロックを示し、1  
-3-1, 1-3-2は別のファイルを構成するブロッ  
クを示す。

【0018】あるファイルの先頭ブロック1-1-1が  
図2に示す位置に配置されたとすると、それに引き続く  
ブロック1-1-2, 1-1-3, 1-1-4は図2に  
示すように先頭ブロック1-1-1の直後に連続して配  
置される。その次のブロック1-1-5はブロック1-  
1-4の直後が塞がっているため、図1に示した方式に  
より配置される。その結果、例えば図2に示す位置に配  
置される。それ以後のブロック1-1-6, 1-1-  
7, 1-1-8は、図2に示すようにブロック1-1-  
5の直後に連続して配置される。

【0019】次に、本発明の具体的な実現方法について  
説明する。

【0020】図4は、本発明を実現するにあたり必要と  
なる情報テーブルの一例である。サーチポインタは図1  
の説明にもあるように、直前にアロケーションされたブ  
ロックの次にアロケーションされるべきブロックを指し  
示し、初期化時、及びファイル削除時に1/2にリセッ  
トされる。具体的には、図4に示すように分母5と分子  
6に分けて保持する。9はランダムアクセス記憶装置中  
に含まれる全ブロック数を示す。10はランダムアクセ  
ス記憶装置に含まれるブロックのうち、どのブロック  
が空きでどのブロックが使用中かを示す情報を保持する  
テーブルである。

【0021】次に、ファイルの先頭ブロックを配置する  
アルゴリズムを示すPADを図4に示す。

【0022】まず、7-1でサーチポインタの分母5が  
現在の値から全ブロック数9を越えるまで7-2~7-  
9を繰り返す。次に、7-2でサーチポインタの分子6  
が現在の値から現在のサーチポインタの分母5を越える  
まで7-3~7-7を繰り返す。7-3では第(サーチ  
ポインタの分子6/サーチポインタの分母5×全ブロッ  
ク数9)番目のブロックが空きかどうかを空きブロック  
管理テーブル10を見て判定し、空きならば7-4~7-  
6を実行する。7-4ではサーチポインタの分子6に  
2を加える。7-5では、7-3で判定したブロックに  
対応する空きブロック管理テーブル10のエントリを使  
用中にし、7-6では7-3で得たブロックの番号を返

してリターンする。7-3で判定したブロックが空きで  
なかったとき、7-7を実行する。7-7ではサーチポ  
インタの分子6を二つ増加する。7-2の判定条件が偽  
になったら7-8, 7-9を実行する。7-8ではサ  
ーチポインタの分子6を1にし、7-9ではサーチポ  
インタの分母5を二倍にしている。7-1の判定条件が偽に  
なったら7-10を実行する。7-10では空きブロッ  
ク管理テーブル10中を先頭から一つずつ空き領域を探  
し、見つけた空きブロックに対応するエントリを使用  
中にし、見つけた空きブロックの番号を返す。

【0023】次に、ファイルの二番目以降のブロックを  
配置するアルゴリズムを示すPADを図6に示す。

【0024】まず、8-1でファイルの最後部のブロッ  
クの次のブロックに対応する、空きブロック管理テー  
ブル10のエントリをチェックする。8-2で、そのブ  
ロックが空きならば8-3, 8-4を、そうでないならば  
8-5を実行する。8-3では8-1でチェックしたエ  
ントリを使用中にし、8-4では8-1でチェックした  
エントリに対応するブロック番号を返してリターンす  
る。8-5では図4で示した処理を行う。

【0025】以上で本発明の一実施例の説明を終わる。

【0026】

【発明の効果】ランダムアクセス記憶装置のデータ領域  
を次々と二分割してファイルに割り当てているため、フ  
ァイルの大きさが小さい場合、かなり多くのファイルに  
連続領域を割り当てることができる。また、サイズの  
大きいファイルもいくつかの連続領域に割り当てられる  
ので、ファイルI/Oの回数を削減できる。記憶装置中の  
ファイルの数が少ないとき、サイズの大きいファイルに  
も一つの連続領域を割り当てることができ、柔軟性にも  
富んでいる。

【図面の簡単な説明】

【図1】ファイルの先頭ブロックの配置方式を示した図  
である。

【図2】ファイルの二番目以降のブロックの配置方式  
を示した図である。

【図3】従来のファイルのブロックの配置方式を示した  
図である。

【図4】本発明を実現するために必要な情報を含むテー  
ブルの一例を示す図である。

【図5】ファイルの先頭ブロックを配置するアルゴリ  
ズムを示したPADである。

【図6】ファイルの二番目以降のブロックを配置するア  
ルゴリズムを示したPADである。

【符号の説明】

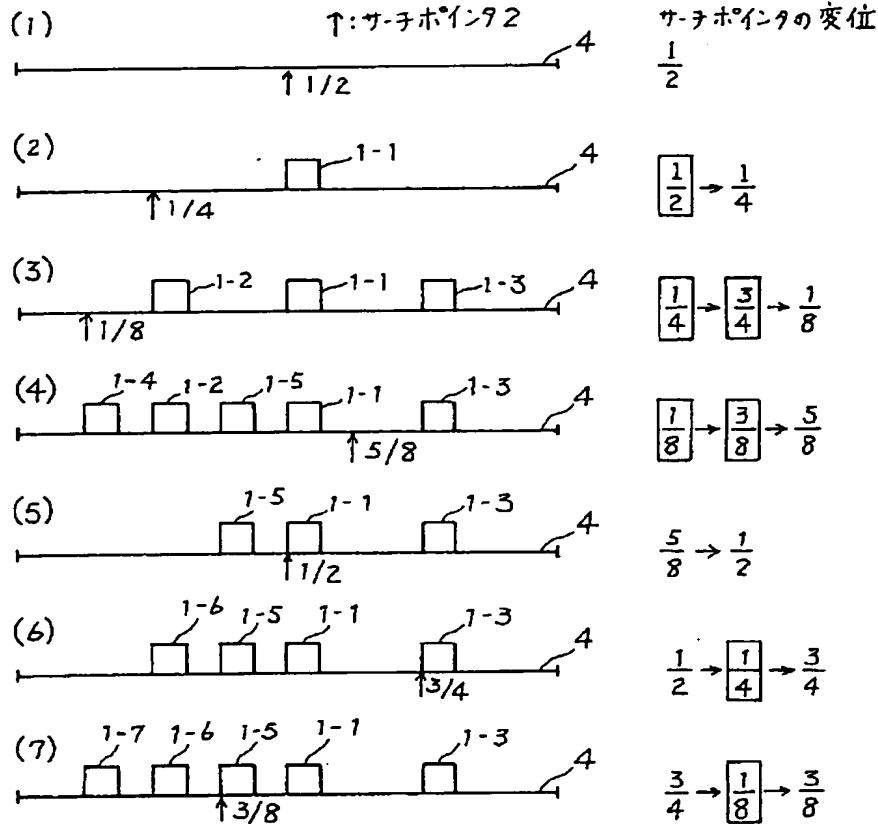
1-1~1-7…ファイルの先頭ブロック、2…サーチ  
ポインタ、3-1~3-4…連続領域、4…ランダムア  
クセス記憶装置のデータ領域、5…サーチポインタの分  
母、6…サーチポインタの分子、7-1~7-9…ファ  
イルの先頭ブロックを配置するアルゴリズムを示したP

AD、8-1~8-5…ファイルの二番目以降のブロックを配置するアルゴリズムを示したPAD、9…ランダムアクセス記憶装置中の全ブロック数、10…ランダム\*

\*アクセス記憶装置中のどのブロックが空きでどのブロックが使用中であることを示す情報を含むテーブル。

【図1】

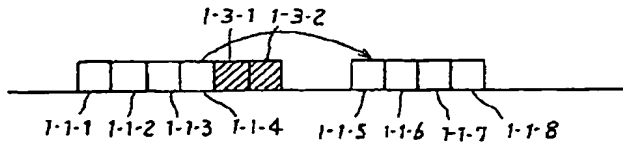
図 1



$\boxed{\phantom{0}}$  は、ブロックがロケーションされたことを示す。

【図2】

図 2



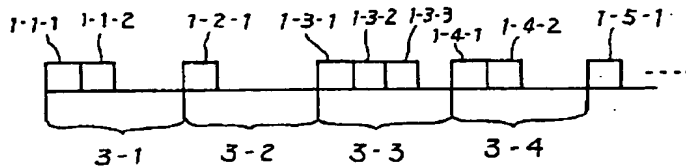
【図4】

図 4

SP1	サ-チポイントの分母5
SP2	サ-チポイントの分子6
全ブロック数	9
空きブロック 管理テーブル	10

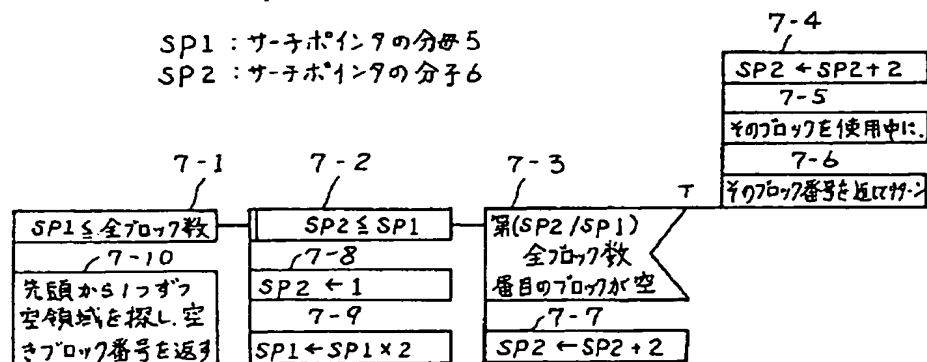
【図3】

図 3



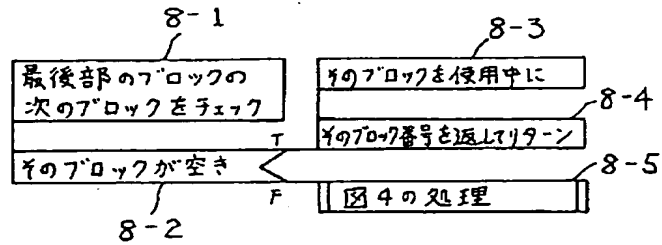
【図5】

図 5



【図6】

図 6




---

フロントページの続き

(72)発明者 岩田 吉隆  
 横浜市戸塚区戸塚町216番地株式会社日立  
 製作所情報通信事業部内